Title: **Optimal Internet Ad Placement**

Inventors: **John B. Ferber and Scott Ferber**

RELATIONSHIP TO PRIOR APPLICATIONS

This application claims the benefit of U.S Provisional Application No. 60/164,253, titled "Optimal Internet Ad Placement Technology," filed November 8, 1999.

BACKGROUND OF THE INVENTION

This invention relates generally to the allocation (e.g. as in a market or exchange) of the supply of a class of products / services with the demand for a class of products / services in an optimal manner (i.e. system-wide best solution since the values of different allocation strategies may vary significantly) that quantifies and accounts for the uncertainty surrounding the supply and demand of different products / services. More particularly, the present invention comprises a system and method for the optimal placement of ads on Web pages.

Optimal ad placement has become a critical competitive advantage in the Internet advertising business. Consumers are spending an ever-increasing amount of time online looking for information. The information, provided by Internet content providers, is viewed on a page-by-page basis. Each page can contain written and graphical information as well as one or more ads. Key advantages of the Internet, relative to other information media, are that each page can be customized to fit a customer profile and ads can contain links to other Internet pages. Thus, ads can be directly targeted at different

customer segments and the ads themselves are direct connections to well-designed Internet pages. Although the present example has been described with respect to traditional Web browsing on a Web page, the same principals apply for any content, including information or messages, as well as advertisements, delivered over any Internet enabled distribution channel, such as via e-mail, wireless devices (including, but not limited to phones, pagers, PDAs, desktop displays, and digital billboards), or other media, such as ATM terminals.

Therefore, as used herein, the term "ad" is also meant to include any content, including information or messages, as well as advertisements, such as, but not limited to, Web banners, product offerings, special non-commercial or commercial messages, or any other sort of displayed or audio information.

The terms "Web page," "Web site," and "site" are meant to include any sort of information display or presentation over an Internet enabled distribution channel that may have customizable areas (including the entire area) and may be visual, audio, or both. They may be segmented and or customized by factors such as time and location. The term "Internet browser" is any means that decodes and displays the above-defined Web pages or sites, whether by software, hardware, or utility, including diverse means not typically considered as a browser, such as games.

The term "Internet" is meant to include all TCP/IP based communication channels, without limitation to any particular communication protocol or channel, including, but not limited to, e-mail, News via NNTP, and the WWW via HTTP and WAP (using, e.g., HTML, DHTML, XHTML, XML, SGML, VRML, ASP, CGI, CSS,

SSI, Flash, Java, JavsScript, Perl, Python, Rexx, SMIL, Tcl, VBScript, HDML, WML, WMLScript, etc.).

The term "customer" or "user" refers to any consumer, viewer, or visitor of the above-defined Web pages or sites and can also refer to the aggregation of individual customers into certain groupings. "Clicks" and "click-thru-rate" or "CTR" refers to any sort of definable, trackable, and/or measurable action or response that can occur via the Internet and can include any desired action or reasonable measure of performance activity by the customer, including, but not limited to, mouse clicks, impressions delivered, sales generated, and conversions from visitors to buyers. Additionally, references to customers "viewing" ads is meant to include any presentation, whether visual, aural, or a combination thereof.

The term "revenue" refers to any meaningful measure of value, including, but not limited to, revenue, profits, expenses, customer lifetime value, and net present value (NPV).

The Internet ad placement technology of the present invention provides an optimal strategic framework for selecting which ad a customer will view next. It maximizes the overall expected ad placement revenue (or any other measure of value), trading off the desire for learning with revenue generation. The technology can be executed in "real-time" and updates the strategy space for every customer.

At its core, the problem is to place the right ad to the right customer. Ad placements are compensated based on the number of successful responses that they generate. This usually means that compensation occurs every time a customer responds

to (e.g., clicks) an ad. Customers respond to ads according to their interests and demands. Thus, a key necessity is to obtain a reliable characteristic profile of each customer. Only with given information about the customer can ads be provided that are targeted towards each customer. Second, there is a need to estimate how different customers will react to different ads. That is, a customer-ad response relation is required. Finally, there is a need for an ad placement technology that optimally decides which ad to show. At the instant a customer opens a page, it is necessary to place an ad. The ad placement technology must incorporate the customer's likely response to each ad and the financial gains resulting from a customer's selection of an ad.

A successful ad placement technology must overcome several critical complications. First, the ad placement algorithm must be sufficiently fast to ensure "real-time" placement. Second, a key element of the technology is its ability to learn through continuous updating. Little information is available about new ads. However, as ads are placed, it can be learned how they relate to various customer profiles. Thus, the technology should both be able to learn and trade off learning versus revenue generation. Finally, the ad placement technology must be able to detect ineffective ads and incorporate minimum and maximum ad placement and ad selection constraints.

BRIEF SUMMARY OF THE INVENTION

This invention concerns optimal ad selection for Internet-delivered ads, such as for Web pages, by selecting and updating an attribute set, obtaining and updating an ad-attribute profile, and optimally choosing the next ad. The present invention associates a

set of attributes with each customer. The attributes reflect the customers' interests and they incorporate the characteristics that impact ad selection. Similarly, the present invention associates with each ad an ad-attribute profile in order to calculate a customer's estimated ad selection probability and measure the uncertainty in that estimate. An ad selection algorithm optimally selects which ad to show based on the click probability estimates and the uncertainties regarding these estimates.

It is therefore an object of the present invention to integrate the optimization and scheduling of web-based ad serving.

It is another object of the present invention to provide an optimal strategic framework for selecting which ad a customer will view next.

It is also an object of the present invention to maximize the overall expected ad placement revenue (or any other measure of value), trading off the desire for learning with revenue generation.

It is another object of the present invention to place ads on Web sites in such a way as to maximize the overall value for the ad serving entity, whether based on impressions, clicks, conversions, or combinations thereof.

It is an object of the present invention to provide an ad placement algorithm that is sufficiently fast to ensure "real-time" ad placement.

It is an object of the present invention to provide an ad placement technology that has the ability to learn through continuous updating.

It is another object of the present invention to provide an ad placement technology

that is able to detect ineffective ads and incorporate minimum and maximum ad placement and ad selection constraints.

It is an object of the present invention to provide an estimate of the probability a customer will click an ad by estimating a principal component vector as well as the ad's click probabilities.

It is yet another object of the present invention to provide binomial updating of click probabilities using principal components, as well as category restrictions and ad blocking.

It is yet another object of the present invention to provide automatic clustering of Web pages in a manner that effectively improves overall Click-Thru-Rates.

It is another object of the present invention to provide optimal delivery of content, messages, and/or ads to customers by any Internet enabled distribution channel.

It is a final object of the present invention to optimize ad placement across a diverse set of media, such as banners, e-mail, and wireless, in an integrated manner via an allocator.

These and other objectives of the present invention will become apparent from a review of the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates the possible use of the present invention in a prior art direct marketing system.

Figure 2 illustrates a first embodiment of the present invention for brand name and mass appeal products.

Figure 3 illustrates a second embodiment of the present invention for lots and niche products.

Figure 4 illustrates a schematic of the present invention.

Figure 5 illustrates the Integrated Channel Management system of the present invention.

Figure 6 illustrates a schematic of the system of the present invention.

Figure 7 illustrates a schematic of the process of the present invention.

Figure 8 illustrates a matching of supply and demand for advertising on Internet enabled distribution channels.

DETAILED DESCRIPTION OF THE INVENTION

The present invention comprises a system and method of optimal ad placement. This invention divides the optimal ad selection problem into three parts: (1) how to select and update the attribute set, (2) how to obtain and update the ad-attribute profile, and (3) how to optimally choose the next ad. For purposes of this description, the application of

the present invention will be illustrated with respect to reconciling the supply of Web pages with the demand for ads on those Web pages in an optimal manner that maximizes revenue. It is assumed that each Web page can only promote one ad at a time, although that is not a limitation of the present invention. Furthermore, the ad provider pays on a per click (ad selection) basis. A typical employment of the invention is illustrated in **figure 1**, wherein customer and client (ad) data **110** is input, turned into information **120** for modeling and used for ad serving **130**, as illustrated in **figure 8**.

The present invention associates a set of attributes with each customer. The attributes reflect the customers' interests and they incorporate the characteristics that impact ad selection.

Similarly, the present invention associates with each ad an ad-attribute profile. The ad-attribute profile has two uses, to calculate a customer's estimated ad selection probability, and to measure the uncertainty in that estimate.

The ad selection algorithm optimally selects which ad to show based on the click probability estimates and the uncertainties regarding these estimates. That is, it optimally trades off current revenue potential with future revenue potential represented by the uncertainty surrounding these estimates. Ads that have been frequently placed will have a well-documented current revenue potential while new ads with few placements represent the possibility of high future potential.

As customers have long-term interests as well as short-term demands the present invention divides attributes into a long-term and a short-term attribute sets. The long-term attribute set measures how much time customers spend in different interest

categories such as business, sports, and health. The short-term attributes detect when a customer is searching for specific products.

Long-term attributes

Long-term customer attributes in the present invention are updated, depending on time and network constraints, on a placement-by-placement or on a time period-by-time period (for example day-by-day) basis. The attributes measure, for example, how much time on a percentage basis a customer spends in each interest group (i.e., sports, gardening, etc.). Thus, suppose that the customer chooses sports half the time and finance half the time. Then sports and finance attributes are each 50% and the remaining attributes are 0%.

Customer interests also change. To accommodate this factor the present invention implements either a moving average or an exponentially-weighted approach to updating each customer's long term attributes. Both of these statistical methods put more emphasis on recent information and can be updated easily.

The attributes together cover all the distinctive characteristics of the customers. There are two ways the attributes are structured. The present invention has a common set of attributes that are always updated. Alternatively, the present invention has two sets of attributes, a base set given by easily available data, and a second set of attributes that are revealed as the customer carries out certain actions.

Short-term attributes

The short-term attribute set of the present invention signals every time there is a specific interest for a particular service or product. For example, suppose a customer is

currently shopping for a computer. Such an event can be detected by specifically marking sites that perform computer comparison tests. The probability that the customer selects a computer ad will be high.

Ad-attribute profiles

Customers also respond differently to different ads. The ad-attribute profile of the present invention measures how sensitive the ad is to the various attributes and thus how likely it is that a customer will react when shown an ad. As the profile for a given customer is not known ahead of time, it must be estimated. This profile estimation algorithm provides an efficient means for updating the attribute estimates in "real time." It is not necessary to store the complete history of customers' responses, but only a set of sufficient statistics for each ad. The sufficient statistics are one square matrix variable with dimension equal to the number of attributes, one vector variable with dimension equal to the number of attributes, and two scalars. Furthermore, the sufficient statistics can be quickly calculated.

The profile estimation algorithm also records the uncertainty of each ad-attribute. The uncertainty conceals an ad's effectiveness (as measured by the true click probability). As an ad's effectiveness directly drives the revenue generation it is important to quickly derive a good estimate. The uncertainty regarding an ad's effectiveness decreases as the number of times it is shown increases.

Optimal Selection

The ad selector of the present invention places ads in a way that maximizes the expected overall long-term ad placement revenue (or any other measure of value). The ad

placement revenue is the compensation received every time an ad is clicked. For the moment, suppose that it is known with certainty the ad-attribute profile for each ad. This means that the probabilities that the customer will react to the ads can be calculated. Multiplying the probabilities with the compensations of the corresponding ads yield the expected ad placement revenues for all ads. The choice that maximizes the expected overall ad placement revenue is then simply the ad with the highest expected ad placement revenue (or any other measure of value).

Unfortunately, one does not know with certainty the ad attribute profiles. This means that the above selection algorithm, if employed using the estimated ad-attribute profile, would not correctly account for revenue generation opportunities of those ads that have not been shown, because it would not incorporate the huge estimation uncertainty of those ads.

This ad-placement algorithm incorporates the uncertainty as well as the expected ad revenue in the selection criterion. Conceptually, the uncertainty is a reflection of the ad's potential upside. That is, it is more likely that the probability of an ad with high uncertainty is significantly higher than its' estimated value than an ad with low uncertainty. Only by testing can the present invention determine whether it is actually true. If true it is clear that there is much to gain in the future.

The ad-placement selection rule works by for each ad combining the volatility and the expected value of the ad placement revenue in a certain way. This rule is based on a dynamic programming approach. This approach yields the true optimal selection algorithm among all possible non-anticipating selection algorithms. The present

invention adapts the dynamic programming solution to obtain a strategy that can be updated in real-time.

The basic modeling technique of the present invention is outlined below and illustrated in **Figure 7**.

Basic Modeling

There are L customers **700** for each of whom the present invention tracks the value of MA customer attributes **702**. Customer attributes 702 may be time-based, geography based, or any other segmentable and tractable attribute. There are N different ads in campaign **704**.

The present invention maintains a *customer matrix*:

Customer ID	Attribute 1	Attribute 2	...	Attribute MA
ID_1	A_11	A_12	...	A_1MA
ID_2	A_21	A_22	...	A_2MA
ID_L	A_L1	A_L2	...	A_LMA

And an *ad matrix*:

Ad ID	Attribute 1 weight	...	Attribute MA weight
Ad_1	W_11	...	W_1MA
Ad_2	W_21	...	W_2MA
Ad_N	W_N1	...	W_NMA

Approach 1

1. The estimated probability of customer x clicking on ad i is given by

$$\sum_{k=1}^{MA} (A\_xk)(W\_ik).$$

2. Every time a customer visits a Web site within the network, the data is collected **712** and the attributes of that customer are updated **714**.
3. Every time a customer is shown an ad, the attribute weightings for that ad are updated **716** depending on how the customer responded.

The calculation of which ad to show **710** is then clearly quick to compute as it is essentially (MA)(N) multiplications and additions and then a comparison of the determined probabilities **708**. With some careful thought, the updates of the customer and ad matrices can also be done rapidly and with numerical stability.

As the present invention collects more data, this method continues to refine the estimates and thus is referred to as Bayesian. Ads may lose their effectiveness over time, and people's attributes will certainly evolve over time. To capture this there are several updating methods that weight recent data more heavily. All of these methods can be updated quickly and require little storage.

In use, as shown in **figures 2** and **3**, a customer accesses a participating Web site at illustrated **201, 301**, an ad server determines the best ad to place (highest score of 150) at **202, 302**, the ad is served to the Web site at **203, 303** and a click by the customer directes him to the advertisers Web site at **204, 304**.

Adding Uncertainty and Optimizing for Earning vs. Learning

Intuitively, there is a big difference between an ad that has been shown 100 times

and been selected once and an ad that has been shown 10,000 times and been selected 100 times, even though each has been selected 1% of the times it has been shown. It is somehow worth something to us to learn more about the first ad, as it is quite possible that it will turn out to be a very popular ad.

The present invention alters the above structure by carrying not just the mean but the standard deviation of each estimated random variable as well.

The ad selection process then works by combining the estimated probability and the standard deviation in a certain way for each ad and then comparing. When done properly, this is the optimal way to balance earning and learning.

Updates of the standard deviation can be calculated quickly as they can be based on the updates of the estimated probabilities.

Adding Structure to the Matrices

The present invention is also able to learn more about a given customer from other customers than the above is yet capturing. As a simple example, imagine that one has discovered that a particular ad is very popular with males and this system is considering showing it to a particular customer. The present invention has an attribute for gender, but doesn't yet know if this particular customer is male or female. However, there is lots of other data about the customer, such as interest level in sports. By looking at the attributes of all *other* customers, and the associated correlations, the present invention can estimate the probability that *this* customer is male. The present invention may find, for instance, that interest in sports is highly indicative of being male.

Choosing the Attributes

A key aspect of the present invention is identifying attributes that are predictive of

behavior. This step requires analyzing real data, and should be re-visited periodically. Second, for numerical stability, the present invention must choose attributes that are not too similar to one another. There are several ways to choose a representative attribute set, basically by selecting orthogonal attributes. Third, the present invention needs concrete policies for deleting non-helpful attributes and splitting ones that are particularly useful. Finally, there are several statistical/data-analysis methods the present invention can employ to create updating procedures for the values of each attribute. The right procedure will depend on initial statistical tests and is also a step that should be re-visited at a later stage.

As customers have long-term interests as well as short-term demands the present invention divides attributes into a long-term and a short-term attribute sets. The long-term attribute set measures how much time customers spend in different interest categories such as business, sports, and health. Thus, suppose that the customer chooses sports half the time and finance half the time. Then sports and finance attributes are each 50% and the remaining attributes are 0%.

The short-term attributes detect when a customer is searching for specific products. For example, a customer shopping for a new computer will likely visit sites that relate to computer sales. Such sites can be marked and computer ads placed on such sites have high probabilities of being selected, while general interest ads have markedly lower probability of being selected.

Searching among the short-term attributes, for ads to show, will be quick as they only flag high probability events.

Advanced Modeling with Integrated Optimization and Scheduling

Every Web site used with the present invention sends a request for an ad every time a user accesses the site. The request is sent to the ad manager. The ad manager has a lookup table specifying ads and associated probabilities defining the ads that should be shown next for every site. This lookup table is updated frequently, such as every hour or on any other relevant time unit basis.

The system records that the ad has been shown and whether or not there was a click. The system holds a database with the number of impressions and clicks for each ad on each site by hour. The system also maintains a list of the total and remaining paid clicks for each ad, and a list of payments per click for each ad.

Basics

The goal of the optimizer-scheduler is to place ads on Web sites in such a way as to maximize the overall value for the advertising serving entity. This value may be a combination of impression, clicks, conversions, and other value that may be obtained by placing an ad on a particular site. The probability of a given ad being clicked on varies from site to site. The present invention does not know these probabilities beforehand but, rather, the present invention continuously refines this estimate as more observations are made. There is value in obtaining additional information about these probabilities and this is accounted for in the algorithm.

Arrangements with Web sites tend to be fairly long-term. Arrangements with advertisers tend to be composed of campaigns, each lasting from days to weeks. The advertisers typically purchase a certain number of clicks. While not always spelled out explicitly, the understanding is that these clicks will occur reasonably uniformly over the

campaign's lifetime. Of course, there is no way to guarantee that an ad does not fall behind schedule (it is possible that nobody chooses to click on the ad). The present invention can, however, ensure (assuming that there is a reasonably rich set of ads) that no ad gets significantly ahead of schedule. This is captured via a tunable parameter within the algorithm.

Occasionally, the arrangement with the advertiser is simply to show the ad a specified number of times. The system of the present invention serves the requested ad according to attributes described above while simultaneously tracking the number of times the ad is displayed.

While taking the full lifetime of each campaign into account, the algorithm explicitly plans for the next 24 hours or other such reasonable period, and then re-optimizes more frequently, such as every hour.

Definitions

System Variables

m denotes the number of Web sites or any reasonable partition of the Web sites in the network.

n denotes the number of ad campaigns or any reasonable collection of ads currently underway.

K denotes the set of ads that are on a pay-per-click basis or any other similar measure of performance.

M denotes the set of ads that are on a pay-per-view basis or any other reasonable measure of activity that is not performance related.

d_j denotes the estimated number of impressions for a first period, such as one 24-hour period or other reasonable period, at site *j*.

μ_j denotes the average clicking probability at site *j* calculated over a second, longer period, such as the past 30 days or other such reasonable period. Only incorporating the observed probabilities for ads that have at least, for example, 500 impressions at that site, then one possible embodiment would be to set $\mu_j = 0.005$ if site *j* is new. Else

$$\mu_j = \max\left(\underset{i;n_{i,j}>500}{\text{Average}}(p_{i,j}), 0.001\right)$$

In this example, the use of 30 days, 500 impressions, and the tolerances of 0.005 and 0.001 are merely exemplary and are not meant as a limitation on the average clicking probability μ_j. Other timelines and constants could also be used without departing from the scope of the invention.

Campaign variables

T_i denotes the total duration in days of ad campaign *i*.

t_i denotes the time in days since the ad campaign of ad *i* began.

C_i denotes the maximum total number of paid clicks for ad *i* over the duration of the ad campaign.

c_i denotes the maximum number of remaining paid clicks for ad *i*.

Π_i denotes the total minimum number of impressions required by ad i over the duration of its campaign.

I_i denotes the minimum number of remaining impressions required for ad i. I_i is updated frequently, such as every hour on the hour.

s_i denotes the payment per click, per view, per conversion, or per any other reasonable measure of activity or performance, depending on the arrangement for ad i.

$n_{i,j}$ is 2 plus the number of impressions for ad i at site j over the last 30 days or other such reasonable period. If the ad has never been shown at site j then $n_{i,j}=2$. (The present invention adds 2 to avoid problems associated with $n_{i,j}=0$)

$k_{i,j}$ is the number of clicks for ad i at site j over the duration of ad i's ad campaign.

$p_{i,j}$ is the observed clicking probability of ad i at site j. If ad i has never been shown ($n_{i,j}=2$) on site j then $p_{i,j}=\mu_j$. Otherwise, $p_{i,j}=\frac{k_{i,j}}{n_{i,j}}+\mu_j\frac{2}{n_{i,j}}$. The second term here is to ensure that the present invention never has $p_{i,j}=0$.

δ_i controls the smoothness of the campaign. This can depend on the smoothness type, how the campaign is doing in terms of delivery, and other factors. A typical value is 0.2. This controls how smoothly clicks must occur throughout the lifetime of a campaign. A value of 0.2 means that no campaign can ever be more than 20% ahead of absolutely smooth (measured daily) delivery.

Parameters

Set γ = 1.5 or any other reasonable number. This is the learning parameter, it controls how heavily the present invention emphasizes learning about ad-site combinations for which the present invention has little information. This will be tuned via simulation.

$\alpha_{i,j}$ denotes the fraction of times ad i should be shown on site j for the next period, such as per hour.

Hourly or Frequent Events

The system sends the number of impressions and the number of clicks for each ad at each site to the ad manager.

The ad manager updates $n_{i,j}$, $k_{i,j}$, and t_i.

The ad manager calculates $p_{i,j}$.

Updating of c_i and I_i

These variables are used in the optimization/scheduling algorithm. First, consider c_i. The contract for most ads specifies the beginning and end of the ad campaign and the maximum number of paid clicks. The scheduling algorithm requires a number that is to be used for one day.

In the formula below, the present invention computes the value of c_i that corresponds to a perfectly smooth delivery of clicks from the current time on. Note that in the linear program (LP), the present invention will not require that this be hit exactly, but rather within a pre-set tolerance.

$$c_i = \frac{\max\left(\left(C_i - \sum_{j=1}^{m} k_{i,j}\right), 0\right)}{\max\left((T_i - t_i), \frac{1}{24}\right)}$$

Now, consider I_i. Sometimes, it is agreed that ad i must obtain a minimum number of impressions. This minimum number must be satisfied at the end of the campaign. As

above, the formula above determines the number of impressions needed during the next day to achieve a smooth delivery of, in this case, impressions.

$$I_i = \frac{\max\left(\left(\Pi_i - \sum_{j=1}^{m} n_{i,j} + 2 * m\right), 0\right)}{\max\left(\left(T_i - t_i\right), \frac{1}{24}\right)}$$

Note that the present invention needs the term $2 * m$ to compensate for the fact the present invention has adjusted n_{ij}.

Scheduling problem (solved frequently, such as once every hour on the hour)

Step 1. Define:

$$\hat{p}_{i,j} = p_{i,j} + \gamma \sqrt{\frac{p_{i,j}\left(1 - p_{i,j}\right)}{n_{i,j} - 1}}$$

Step 2. Solve the following linear programming problem:

$$\underset{\{\alpha_{i,j}\}}{\text{MAX}} \sum_{i \in K} \sum_{j=1}^{m} \alpha_{i,j} v_{i,j} d_j \qquad (1)$$

$$\text{Subject to } \sum_{j=1}^{m} \alpha_{i,j} p_{i,j} d_j \le (1 + \delta_i) c_i, \quad i \in K \qquad (2)$$

$$\sum_{j=1}^{m} \alpha_{i,j} d_j \le (1 + \delta_i) I_i, \quad i \in M \qquad (3)$$

$$\sum_{i=1}^{n} \alpha_{i,j} \le 1, \quad j = 1, 2, \ldots, m \qquad (4)$$

$$\alpha_{i,j} \ge 0, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m \qquad (5)$$

where $v_{i,j} = \hat{p}_{i,j} s_i$ if ad i is click-based or conversion-based, and s_i if it is impression-based.

Comments

(1) The objective function is to maximize the overall value, including learning about sites where we have little information.

(2) The LHS is the total number of expected clicks for ad i during the interval. This constraint enforces the campaign smoothness condition.

(3) The LHS is the total number of expected impressions for ad i during the interval. This constraint enforces the campaign smoothness condition.

(4) This constraint ensures that the probabilities of what ads to show at each site add to 100%.

(5) This constraint ensures that all probabilities are non-negative.

Remarks

(1) By setting $s_i = 1$ for all i converts the objective function into one that seeks to maximize the overall Click-Thru-Rate (CTR).

(2) There is no explicit constraint ensuring that each ad does not fall "too far behind". The reason for this is such a constraint would lead to the linear program (LP) having no feasible solution.

(3) To account for the remark above, campaigns should be monitored on a frequent basis (daily) with poor ads being removed or outsourced.

(4) Note that there is obviously always a solution to the LP.

Creating an Ad lookup table

The present invention describes the process of converting the output of the linear program (LP) into a lookup table. For each site j and ad i multiply the $\alpha_{i,j}$ by 100 and round off the product to the nearest integer. Let $\beta_{i,j} = \text{Round}(100 * \alpha_{i,j})$. $\beta_{i,j}$ represents

how many times out of a hundred ad i should be shown at site j. Create a list for site j by letting the first $\beta_{1,j}$ elements be ad 1, let the next $\beta_{2,j}$ be ad 2, and so forth. This process will yield a list of approximately 100 ads for each site (many ads will appear several times for a given Web site). The next step is to ensure that the list has exactly 100 ads for each site. This is done by truncating the list for any site with more than 100, and repeating the first ad on the list as many times as necessary for any site with less than 100.

It is possible to employ a frequency-capping component at this stage of the algorithm.

Daily routine

Calculate d_j and μ_j over the last 30 days or other such reasonable period, as shown in the schematic diagram of **Figure 4**. When new sites or new ads **410** are added, constraints are prepared **420**, and the new matrices are added to the ad server's optimization engine **430**. Prior to having adequate data, initial estimates (alphas) **435** are used and the data is added to the ad look-up tables **440**. The ads are then served at **450** (with testing **490** and frequency capping **492**). Response data is collected at **460** and recorded together with the ad serving information in transaction log **470**. The data is then used to update parameters at **480**, and the iterative process continues.

Enhancements

This framework allows for a number of additional constraints to be added in a natural way.

Click probability estimation with principal components

Above, the probability that users visiting Web site j will click on ad i was estimated by dividing the number of clicks on ad i at Web site j with the number of impressions of ad i at Web site j, but can be estimated by any other reasonable method.

An alternative is a principal component approach to banner ad probability estimation. This approach contains two steps. In the first step we estimate the principal component vectors whereas in the second step we estimate the banner ads' click probabilities. Each step are updated as new information becomes available. The advantage to using the principal component approach is significant. For example, if there are 100 Web sites and 5 principal components then the conventional approach requires approximately 20 times as many impressions as the principal component approach to reach the same level of accuracy.

This approach is begun by presenting a series of definitions. It continues by describing the principal component estimation, and concludes by finally describing the probability estimation.

Definitions

Probabilities	Estimate of the probability that users downloading ad i from Web site j will click on that ad is $p_{i,j}$.
Error	Uncertainty of the estimate $p_{i,j}$ is $\sigma_{i,j} = p_{i,j} * (1 - p_{i,j})/n$, (a slightly biased estimate),
Sites	There are m sites.
Site Average	Let μ_j denote the average click probability on site j.

Normalized Ad probability Vector - For each ad i we define the vector

$$y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,m}] \text{ where } y_{i,j} = \frac{(p_{i,j} - 1)}{\sigma_{i,j}}.$$

Principal Components - hypothesize that there exist l m-dimensional vectors $x_1, x_2, \ldots, x_l$, such that every ad probability vector is a linear combination of $x_1, x_2, \ldots, x_l$.

Other Let $n_{i,j}$ denote the number of impressions of ad i on Web site j and let $k_{i,j}$ denote the number of clicks of ad i on Web site j.

Principal components estimation

When using principal components estimation, the present invention identifies ads that have been shown a large number of times at many Web sites. These are the ads that will be used to calculate the principal components.

Step 1. Calculate estimation of site averages.

$$\mu_j = \frac{\sum_i p_{i,j}}{\text{Count}(\text{i on j})}$$

Step 2. Calculate the variance of the error of each probability estimate.

$$\sigma_{i,j} = p_{i,j} * (1 - p_{i,j}) / n$$

Step 3. Calculate normalized ad probability vectors.

Step 4. Calculate the principal components by first creating the matrix Y. Row *i* of Y corresponds to ad i. Then calculate the matrix product $Y^T Y$. Then find the eigenvectors and eigenvalues of $Y^T Y$. Choose the *k* eigenvectors corresponding to the *k* eigenvalues which together accounts for at least x% of the total of the sum of all eigenvalues. The first principal component corresponds to the first eigenvector as follows: Element *i* of the

eigenvector is the weight associated with ad i. Therefore, multiply the elements of the first eigenvector with their corresponding estimated probabilities for each site and sum over these newly found values to determine the first principal component vector. Repeat the procedure for the remaining k-1 eigenvectors.

Banner ad click probability estimation

With the principal components available there are a variety of ways to estimate an ad's click probabilities. Two straightforward methods of such estimation are ordinary least squares regression and generalized least squares regression.

The objective of the principal component approach is to efficiently and quickly obtain ad probabilities for a majority of banners. In addition to finding the probabilities for the majority it is also necessary to identify banners where the principal components do not capture a significant portion of the observed probabilities. A maximum likelihood approach can be used to integrate this aspect into the probability estimation routine.

Binomial Updating of Click Probabilities Using Principal Components

Consider a row of n cells that have unknown click probabilities p_i, where cells are $i = 1,2,\ldots,n$

Assume there is a single (for notational simplicity) principal component that is likely to give these probabilities. This principal component is a vector $v = (v_1, v_2, \ldots, v_n) \geq 0$. Then model the vector P as

$$P = av + e$$

where a is an unknown constant and $e = (e_1, e_2, \ldots, e_n)$ is a vector of errors.

Then assume that the e_i's are independent, normal random variables with zero mean and variance σ^2. The variance is determined by the process that determines the principal components.

Now, imagine the system has been run for a while and has observed k_i clicks from n_i impressions in cell i. It is then desirable to assign the best p_i's.

The joint probability of those click rates and the probabilities given a is

$$P = \prod_{i=1}^{n} \exp\left[\frac{-1}{2\sigma^2}(p_i - av_i)^2\right] \ \prod_{i=1}^{n} p_i^{k_i}(1-p_i)^{n_i - k_i} C$$

where C is a constant independent of a and the p_i's.

Now determine a and the p_i's by maximizing P with respect to a and the p_i's. Ignoring C, to obtain:

$$\ln P = \frac{-1}{2\sigma^2}\sum_{i=1}^{n}(p_i - av_i)^2 + \sum_{i=1}^{n}[k_i \ln p_i + (n_i - k_i)\ln(1-p_i)] \qquad (*)$$

Note that $\ln P$ is concave with respect to a and p_i's ≥ 0, so maximization is well-defined.

Note that (as one would expect) if $\sigma >> 0$ and/or n_i, k_i large, one finds $p_i = \frac{k_i}{n_i}$. Also, for σ small and/or n_i, k_i small, one finds $p_i = av_i$.

Now, the problem is separable with respect to p_i's, so one strategy is to maximize with respect to p_i with a fixed. This gives the necessary condition:

$$F(p_i) = \frac{-1}{\sigma^2}(p_i - av_i) + \frac{k_i}{p_i} - \frac{(n_i - k_i)}{(1-p_i)^2} = 0$$

Note that $F(0) = +\infty$ and that $F(1) = -\infty$. Hence, there is a p_i with $0 < p_i < 1$ and $F(p_i) = 0.$

Furthermore,

$$F'(p_i) = \frac{-1}{\sigma^2} - \frac{k_i}{p_i^2} - \frac{(n_i - k_i)}{(1-p_i)^2} < 0$$

so F is monotone. Thus, the solution is unique.

It can therefore be concluded that for a given a, there is for each $i = 1,2,\ldots,n$ a unique p_i, $0 < p_i < 1$, that can be easily found by Newton's method or any other descent method. (The case of $k_i = 0$ is handled separately later.)

Now, consider p_i to be a function of a. Then,

$$\begin{aligned}\frac{\partial}{\partial a}\ln P &= \frac{\partial \ln P}{\partial a} + \sum_{i=1}^{n} \frac{\partial \ln P}{\partial p_i} P_0' \\ &= \frac{\partial \ln P}{\partial a} \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(p_i^{(a)} - av_i)v_i\end{aligned}$$

This discussion motivates the following algorithm:

1. Select initial a
2. Find the p_i's by solving $F_i(p_i, a) = 0$ (Newton's method 1 variable at a time)
3. Evaluate $\frac{\partial}{\partial a}\ln P$
4. Adjust a by steepest descent

Note that the extension to multiple principal components is straightforward.

Case of $k_i = 0$

The necessary condition is

$$(1-p_i)(av_i - p_i) = n_i \sigma^2$$

It is easy to see that if $av_i > n_i \sigma^2$, then there is a solution with $0 < p_i < 1$. Otherwise $p_i = 0$. should be used. Putting this together, $p_i = \max\{root_1, 0\}$ where $root_1$ is the root of the quadratic less than 1. That is,

$$root_1 = \frac{1 + av_i - \sqrt{(1+av_i)^2 - 4(av_i - n_i\sigma^2)}}{2}$$

Note that it follows from this that if $n_i = 0$, we have $p_i = av_i$. If $k_i = 0$ repeatedly, one does not set $p_i = 0$ until they get at least $n_i = \frac{av_i}{\sigma^2}$ impressions.

Initial value of a

If all the n_i's are small, and/or σ^2 is small, we set $p_i = av_i$ for all i.

Then,

$$\mathrm{LnP} = \sum_{i=1}^{n} k_i \ln av_i + \sum_{i=1}^{n} (n_i - k_i) \ln(1 - av_i)$$

$$\frac{\partial \ln P}{\partial a} = \sum_{i=1}^{n} \frac{k_i}{a} - \sum_{i=1}^{n} \frac{(n_i - k_i)}{1 - av_i} v_i = 0$$

Solve for a.

This can be interpreted by multiplying by a.

$$\sum_{i=1}^{n} k_i = \sum_{i=1}^{n} (n_i - k_i) \frac{av_i}{1 - av_i}$$

which shows that a is set to balance the overall probabilities consistent with observed clicks and impressions.

Prior distribution on a

Adding a prior density on a as

$$\frac{1}{\sqrt{2\pi\omega}}\exp\{-\frac{1}{2\omega^2}(a-a_0)^2\}$$

This adds the term

$$-\frac{1}{2\omega^2}(a-a_0)^2$$

to lnP as defined in (*) above.

Category restrictions

Certain advertisers would like to have their ads displayed only on a subset of the sites. This is handled in the following way. Let the subset of such sites be denoted by *J*. This might be, for example, the set of all sports related sites. Then, if the present invention is considering ad *i*, the restriction takes the form:

$$\alpha_{i,j} = 0 \text{ for all } j \notin J.$$

The subset *J* can, of course, involve multiple levels of categories, generally chosen by the advertiser. A typical subset could be something like 'all of the sports related – Spanish language – G-rated sites.'

Ad Blocking

Conversely, certain Web sites would like to prevent particular ads from appearing on their site. This may be the case, for instance, if the item being advertised is viewed as

a competitor to the Web site's product. Let the site be denoted by j and the set of ads to be blocked to be denoted by the set I. Then the restriction has the form

$$\alpha_{i,j} = 0 \text{ for all } i \in I.$$

Typically, a Web site would be able to do this by both blocking entire categories, such as R-rated sites, and by selecting particular ads for exclusion, such as one of a direct competitor.

Click-Thru-Rate (CTR) of impression based ads

Even with contracts that are strictly impression based, it may be advantageous to attempt to enhance the CTR of such ads. Providing a good CTR may lead to more future business. To do this, the present invention must determine how valuable each click on an impression based ad is in economic terms. Then, this can simply be added to the objective function.

Clustering process

Automatic clustering of small Web sites can be employed in a manner that effectively improves overall Click-Thru-Rates. To form clusters, the process starts by matching each ad with a *campaign type*, which is assigned through a GUI. There are types for 'Personal Finance', 'Sports', 'Computers and Technology', and the like. The present invention denotes each campaign type $t_i, i = 1,2,...,20$, and the set of all campaign types $\mathbf{T}$. Each cluster will correspond to one of these types.

To determine which types will be used for clustering, a database is used with the history of the last 30 days or other reasonable period, and count all the impressions for each type. If the objective is to form *n* clusters, then the first *n* types ordered by

descending number of impressions are selected to be the *clustering types*. Now call each clustering type $\hat{t}_j, j = 1,2,\ldots,n$, and the set of all the clustering types $\hat{\mathbf{T}}$. Each clustering type is assigned a number (ID) starting from 2 and going up until *n+1*. A Webmaster with cluster ID = 0 means that it was not clustered, and with ID = 1 means it is in a cluster of special Webmasters.

The database contains information on all the campaign types that each Webmaster showed. Not all webmasters-type pairs in the database will be used to perform the computations; in one embodiment, only those that meet the following requirements:

- It must have more than 2 impressions on a type
- It must have more than 1 click on a type
- The CTR for a type must be less than 100%

Although this is a preferred screening process, any other such reasonable screening process can be used without departing from the scope of the present invention.

In addition, the set of campaign types for a Webmaster must be a superset of the clustering types: $\hat{\mathbf{T}} \subseteq \mathbf{T}_m$, where m represents a particular Webmaster.

Each Webmaster will be assigned to one and only one cluster, so it will have a corresponding cluster ID, ID_m. Only one more piece of information is needed to determine the cluster ID of each Webmaster: p-hat.

$$p\_hat_{m,\hat{t}_i} = CTR_{m,i} + \gamma \sqrt{\frac{CTR_{m,i}\left(1 - CTR_{m,i}\right)}{imps_{m,i}}},$$

where γ is a learning parameter m is the Webmaster, i is the campaign type, and $imps_{m,i}$ refers to the number of impressions for the Webmaster-campaign type pair. Now,

$$ID_m = 1 + \arg\max_{j}\left(p\_hat_{m,i_j}\right), j = 1,2,\ldots,n.$$

Each j corresponds to a clustering type, as defined before.

Thus, the object is to look for the max p-hat for each Webmaster. The type associated with the max p-hat will be cluster assigned to the Webmaster. In order to write the output, the present invention translates the type to its cluster ID.

Splitting large clusters

It could be the case that once clusters are formed, the total number of impressions for one of them will be over 20% or any other reasonable set percentage of the total number of impressions for all the clusters. In this case, it is desirable to split the cluster by applying the clustering process to those Webmasters in the largest cluster, and by forming a new set of two clustering types for them that excluded the type associated with the cluster. For instance, if cluster 3 with associated type 'Sports' is the target, then a new clustering type set might be {'Entertainment', 'Health'}, which will be chosen because they are the two types with the most and second-most impressions. Each Webmaster will be assigned a new cluster ID using the same "max p-hat" criteria.

The splitting process is repeated until no cluster has more than 20% of all the impressions.

Integrated Channel Management

It is also desirable to optimize ad placement across a diverse set of media, such as banners, e-mail, and wireless, in an integrated manner. An allocator **500**, as shown in **Figure 5**, can be used to serve full-sized **510**, odd-sized **520**, and other type **530** ads using the following algorithm:

Definitions

V_i = Expected impressions per period, such as per day, of media type i.

p_{ij} = probability of a click on media type i for campaign j.

G_j = Total target number of clicks for campaign j for the period.

ς_{ij} = The percent of all impressions from media i that will be allocated to campaign j.

$$\text{Max} \sum_{i,j} p_{ij}\varsigma_{ij}V_i$$

$$\text{s.t.} \quad \sum_j \varsigma_{ij} \le 1 \quad \text{for all } i$$

$$\sum_i p_{ij}\varsigma_{ij}V_i \le (1+\delta)G_j \quad \text{for all } j$$

$$\varsigma_{ij} \ge 0 \quad \text{for all } i \text{ and } j$$

Of course, constraints enforcing minimum and maximum representation on various channels are possible as well.

Then, $p_{ij}\varsigma_{ij}V_i$ is sent to the LP as the upper bound for campaign j for channel type i.

Multiple ads from one customer

From time to time, an advertiser will employ multiple banner designs. One approach to this, of course, is simply to treat each of these as a separate ad. However, if the advertiser is willing to let the optimizer select which ads to show, the present invention can expect on average an improvement in the CTR. Imagine that the two ads are labeled l and m, and that the initial click totals (on an average daily basis) were c_l and c_m. Then, normally the present invention would have included the two constraints:

$$\sum_{j=1}^{m} \alpha_{l,j} p_{l,j} d_j \leq (1+\delta) c_l$$

$$\sum_{j=1}^{m} \alpha_{m,j} p_{m,j} d_j \leq (1+\delta) c_m$$

Instead, the present invention can replace this with the single constraint, which is less restrictive and therefore will result in a better or equal solution:

$$\sum_{j=1}^{m} (\alpha_{l,j} p_{l,j} d_j + \alpha_{m,j} p_{m,j} d_j) \leq (1+\delta)(c_l + c_m)$$

It is also possible to do something in between the above two solutions. For example, an advertiser with two different ad designs could ask for a total of 10,000 clicks with a minimum of 2,500 each. Therefore, there are many other reasonable solutions.

The method of the present invention can be practiced by conventional servers **620, 630**, such as Pentium III based systems operating with Windows NT, interacting over the Internet **600** to collect attribute information about customers **640** and ads from database **610**, and then serve the ads to the customers **640** operating Internet enabled devices with browsers, such as Apple Macintosh or Windows-based personal computers with browser clients like Internet Explorer or Netscape Navigator, as shown in **figure 6**. As such, there are no special requirements for the user interaction on the Internet using the present invention. Conventional PCs, which may be Pentium based or Apple Macintosh type processors, are all suitable processors for exercising the present invention. Likewise, the server of the present invention can be an Intel Pentium type server, Sun server or other server suitable for serving advertisements.

Numerous aspects of the present invention also have separate utility outside of any Internet enabled distribution channels. The basic modeling methodologies and algorithms

of the present invention are therefore able to be incorporated with virtually any other marketing medium in which an "ad" is displayed to a "customer," including, but not limited to, mail, telephone, facsimile, television, radio, and print media. Other embodiments, with modifications and changes to the preferred embodiment, will be apparent to those skilled in the art without departing from the scope of the present invention as disclosed. Therefore, the present invention is only limited by the claims appended hereto.